A High-Level Specification for Adaptive Ecological Momentary Assessment: Real-time Assessment of Drug Craving, Use and Abstinence

Jia-Ling Lin, Ph.D.[1], Massoud Vahabzadeh, Ph.D.[1], Mustapha Mezghanni, M.S.[2],
David H. Epstein, Ph.D. [1], and Kenzie L. Preston, Ph.D. [1]

[1] *DHHS, National Institutes of Health, National Institute on Drug Abuse, Intramural Research Program*
[2] *Johns Hopkins Bayview Medical Center, Baltimore, Maryland*

Abstract

In psychological research, efforts to capture day-to-day human experience traditionally relied on pen-and-paper diaries and questionnaires. Some current studies, however, incorporate handheld computers, which provide researchers with many options and advantages in addition to providing more reliable data. One advantage of using handheld computers is the programmability of the electronic diary, which, compared to old-fashioned paper diaries, affords the researchers with a wealth of possibilities. An important possibility is to construct a built-in mechanism in the computer-administered questionnaires that would allow transparent branching, in which question presentation is contingent on participants' answers to previous questions. The major hurdle in implementing such an approach is the limitations of the platform used for such assessments: inexpensive "low-end" handheld devices. We propose a high-level specification which enables non-programming researchers to "branch" their questionnaires without modifications to the source code in a highly user-friendly fashion, with backtracking capability and very modest hardware requirements. A finite state automaton approach was implemented, we believe for the first time, to create an auto-trigger mechanism for the real-time evaluation of the conditions. This solution provides our investigators with the capacity to administer efficient assessments that are dynamically customized to reflect participants' behaviors without the need for any post-production programming.

Introduction

Handheld computers are undoubtedly a promising research tool that will become even more attractive as the relevant technologies further mature [1]. A central focus of our outpatient clinic is to learn why formerly drug-dependent patients often relapse to compulsive patterns of drug use long after withdrawal symptoms have abated. To study this without the biases introduced by retrospective data collection (in which participants are asked, after the fact, why they relapsed), we have begun to use *electronic diaries* (*ED*s) in which they record where they are and what they are doing several times a day.

It has been shown that EDs are an especially reliable tool for sampling day-to-day experience [2]. One major advantage of using EDs is that participants do not need to rely on their memories and the system prompts them as necessary or allows them to enter an assessment as an event necessitates such action (such as one triggered by a craving event). Electronic-based *Ecological Momentary Assessment* (*EMA*) has the advantage of gathering data on base rates of daily events (through randomly prompted sampling) along with data on specific occurrences (event-contingent sampling). EMA has thus provided valuable insight into such issues as the understanding of marital and family processes within the context of daily life in a way that is not possible with more traditional methods [3].

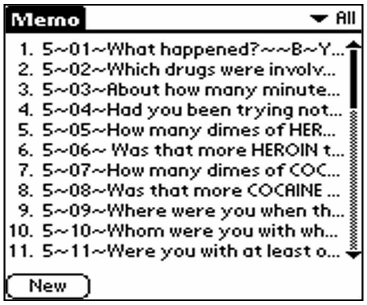
In order to accommodate the economic constraints of the project, the platform had to be inexpensive; hence, low-end *Palm Zire* handhelds were chosen. We determined very early in the project life cycle that the major obstacle would be the hardware limitations of these inexpensive devices, especially in terms of onboard memory and storage. Consequently, the solution had to use the hardware resources of a handheld device in a most frugal and efficient manner.

The *branching* capability was highly challenging to implement with minimal hardware resources, especially since the system required the capacity for the users (e.g., investigators) to be able to create their *branchable* questionnaires post-production without any modifications to the actual code and recompilation of the code. Our prior clinical experience with standard questionnaires has shown that when participants are given branching questions on paper (e.g. "If you did not use a syringe in the past 7 days, skip to question 12"), they frequently become confused and fail to complete the questionnaire correctly. We wanted to avoid this problem in our EMA studies. An even more challenging requirement was the provision of one level of backtracking for the participants, allowing them to modify their prior answers while the system correctly recalculates the new branching positions based on the modified answer without using a large amount of hardware resources.

To address these issues, we developed a high-level specification and implemented a finite state automaton to provide such a solution with the required levels of versatility and adaptability while using the most economical handheld devices. In this paper, we will describe the mechanism which enables the researchers to create questions without modification of the source codes in our handheld *Transactional Electronic Diary (TED)* [4], a highly configurable EMA program with many features that had previously not been available such as the ability to perform true EMA and branching capabilities. We used Metrowerks CodeWarrior DevStudio for Palm OS Platform v9.0 in C language for developing the system.

A High-Level Specification: Creating Assessments to Be Asked

In our system each sequence of questions and the participant's responses represent one trial. The collection of such trial records is stored on the handheld device for subsequent transferring and integration into the participant's clinical/research records conducted by synchronization of the device with designated desktop computers in the clinic.



1. 5~01~What happened?~~B~Y...
2. 5~02~Which drugs were involv...
3. 5~03~About how many minute...
4. 5~04~Had you been trying not...
5. 5~05~How many dimes of HER...
6. 5~06~ Was that more HEROIN t...
7. 5~07~How many dimes of COC...
8. 5~08~Was that more COCAINE ...
9. 5~09~Where were you when th...
10. 5~10~Whom were you with wh...
11. 5~11~Were you with at least o...

Figure 1. Assessment definition file in TED.

TED stores the questions in the Palm's *Memo Pad*. To create an entry in the Memo Pad, one can either use the stylus or the onscreen keyboard to enter text within the Memo Pad, or type into the *Palm Desktop* software, and then transfer the typed data (via *HotSync*) into the Memo Pad. Figure 1 depicts a screen shot of the Memo Pad which partially displays an assessment definition. Memo Pad automatically inserts an index number for each newly created memo entry. Figure 1 shows eleven memopad entries; each entry is a question construct.

It is important to note that the investigator can set TED to take over the handheld computer, refusing to let any other programs run until the experiment is over. This helps prevent participants from disrupting data collection or wasting the batteries; it also reduces or eliminates the apparent resale value of the handheld computer since only users with appropriate credentials can reset the unit. TED requires participants to enter a password before questions are displayed. The details of TED as well as the comparisons with other systems can be found in [4].

Syntax of a Question Construct

The syntax of an assessment question in TED is:

Assessment ID ~ Question ID ~ Question Text~ Condition ~ Question Type ~ If Mandatory Question ~ If With Comment ~ Max Chars ~ {Label 1 ~ Label 2 ~ … ~ Label n}

where "~" is a field separator. Details of each of the fields are given below.

- *Assessment ID*: an integer ID number, either positive or negative (ranging from -127 to 127) that uniquely identifies the assessment. By default, Assessment ID 5 is reserved for the event-contingent sampling, and 6 for the random sampling.
- *Question ID*: an integer ID number, either positive or negative (ranging from -32000 to 32000; 0 excluded) that uniquely identifies the question of the assessment.
- *Question Text*: it may contain any text except the symbol "~".
- *Condition*: a condition to display the question. For example, "*(Q1=2 & Q3≠5) | Q4=2*". This is the field in which the investigator can specify branching requirements.
- *Question Type*: one character from the choices of *T*, *N*, *C*, and *B* which stands for text field, numeric field, check boxes, and buttons, respectively. Note that a button permits one answer only, which is equivalent to a radio button.
- *If Mandatory Question*: *Y* if an answer is required; otherwise *N*.
- *If with Comment*: *Y* if the participant should have a text-entry field after the buttons or the checkboxes; otherwise *N*.
- *Max Chars*: the maximum number of characters and digits allowed for the text/comment and numeric fields, respectively.

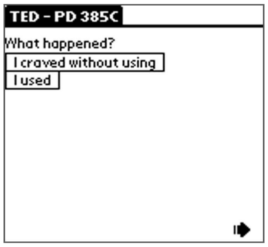
- *{Label 1 ~ Label 2 ~ … ~ Label n}*: labels for the response checkboxes or buttons.

The syntax of a condition will be explained shortly. Note that in case there is no need to specify a field, it can be skipped by leaving the field position empty.

In order to conserve the available memory, TED stores the trial records data that are repetitive in a separate log file and subsequently matches that information to the associated collected data. For example, the timestamps of the beginning and the termination of the trial are stored once per trial in a log file and then using the trial ID, they are linked to the question ID and the answers to the provided questions. One may create as many questions as needed, limited only by available memory.

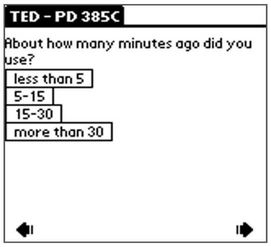
Assessment Questions Examples

The following examples demonstrate the prescribed syntax for the assessment questions and the branching functionality of the system. The EMA application will display Figure 2-a when processing the following non-conditional question: *5~1~What happened?~~B~Y~N~~I craved without using~I used*



What happened?

Figure 2-a. Question 1 of the event-contingent sampling as displayed on a handheld computer.



About how many minutes ago did you use?

Figure 2-b. Question 3 of the event-contingent sampling.

And Figure 2-b is displayed for the following conditional question: *5~3~About how many minutes ago did you use?~Q1=2 ~B~Y~N~~less than 5~5-15~15-30~more than 30*

In the second example, the presentation of the question to a participant will occur if and only if the condition *Q1=2* (which means whether *Q1*, the user answer to question 1, is equivalent to the second option of the answers to question 1) is evaluated to *True*. The real-time evaluation of a condition provides the capacity to administer efficient assessments that are dynamically customized to reflect participants' behaviors with the special feature of one-level backtracking.

Assessment Branching: Adaptive EMA Using Finite Automata

In practice, clinical investigators may initially compile a multitude of questions intended to monitor target behaviors or symptoms in study participants. The issue is how to present the *right* questions to participants in real time. For example, in our current study of craving and relapse in polydrug abusers, the participants might first indicate whether they had merely craved drugs or had actually used them. Then contingent upon having used them, the participants indicate which drug or drugs were used, and then, contingent upon how many different drugs were used, they answer follow-up questions about each drug.

TED system provides a dynamic, auto-trigger approach for efficient questionnaire design. This auto-trigger approach is enabled by a *condition* (a novel feature in our high-level specification) of a question. A question will be displayed if and only if its condition is evaluated to *True*. A condition is a *Boolean* formula with atoms being comparisons between *answers* and constants. An answer to question *i* is denoted by *Qi*. Valid operators are: +, -, *, /, >, <, =, >=, <=, ≠, &, |, (,) and !.

Note that TED system evaluates and displays one question at a time in the order of question IDs. If there is no condition associated with a question, then the question is displayed without exception. Otherwise, *the* condition associated with the question will be evaluated to determine whether to display the question. Once a condition is evaluated, it will not be re-evaluated unless its question is backtracked. No loops will occur in the question flow for there is no *go to* (specifically, jumping back to previous questions) statement in the specification of a condition.

This dynamic approach is implemented based on the theory of *finite state automaton* (*FSA*) [5, 6]. The advantage of using a finite state automata approach is that we can efficiently implement the EMA system within the limited resources of an inexpensive device, such as those available on a low-end Palm Zire handheld. In addition, the use of FSA makes the implementation easier to maintain and, moreover, to extend in the future.

In order to evaluate a condition, we first substitute an answer with its value (as shown in the block of *Substitution* of Figure 3). After preprocessing the answer, we pass the entire expression to our implementation of the FSA for calculating its truth value (as shown in the block of *Process Automata* of

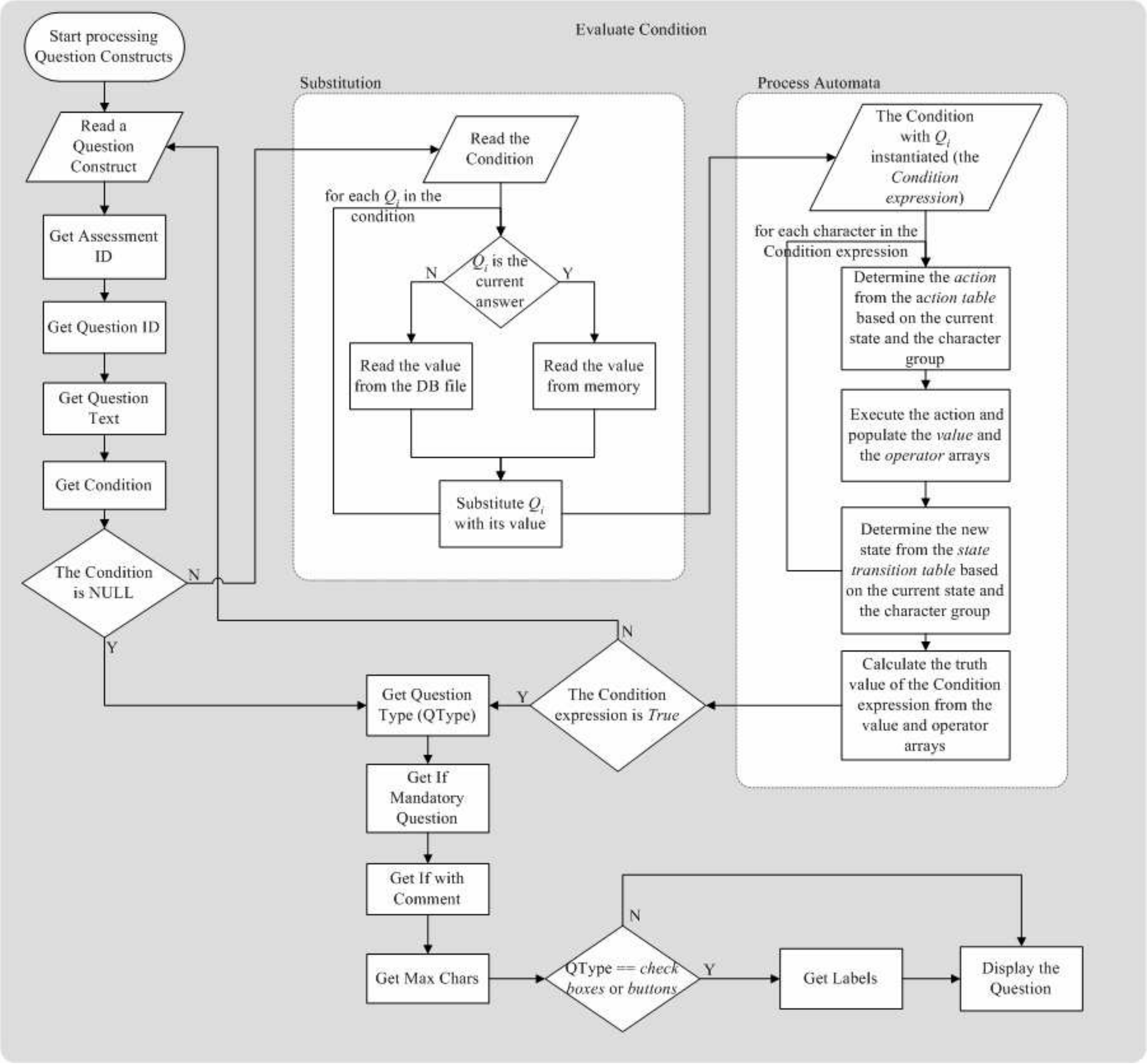

Start processing Question Constructs
Evaluate Condition
Substitution
Process Automata
Read a Question Construct
Get Assessment ID
Get Question ID
Get Question Text
Get Condition
The Condition is NULL
N
Y
Read the Condition
for each Q_i in the condition
Q_i is the current answer
N
Y
Read the value from the DB file
Read the value from memory
Substitute Q_i with its value
The Condition with Q_i instantiated (the *Condition expression*)
for each character in the Condition expression
Determine the *action* from the *action table* based on the current state and the character group
Execute the action and populate the *value* and the *operator* arrays
Determine the new state from the *state transition table* based on the current state and the character group
Calculate the truth value of the Condition expression from the value and operator arrays
N
The Condition expression is *True*
Y
Get Question Type (QType)
Get If Mandatory Question
Get If with Comment
Get Max Chars
QType == *check boxes* or *buttons*
N
Y
Get Labels
Display the Question

Figure 3. The flow diagram demonstrates the process used in our *Transactional Electronic Diary (TED)* for the assessment of the Question Constructs and real-time evaluation of the branching conditions by execution of the finite state automaton.

Figure 3). The output of the FSA is either *True* or *False*. For instance, if a participant chooses the third answer of the first question (i.e., *Q1*), then the value of *Q1* is evaluated to 3. If a condition is, for example, of the form *Q1=2*, then its truth value will be evaluated to *False* since 3 is not equal to 2. Our approach also uses the FSA to parse the indicated condition for syntax validation. Syntax validation and condition evaluation are performed concurrently.

The input alphabet Σ of the FSA is {*blank*, +, -, *, /, >, <, >=, <=, 0, .. 9, ., (,), !, &, |}. The set of states $S = \{S_k \mid k = 0, \ldots, 9\}$, with S_0 being the initial state. The set of final states $F = \{S_9\}$ will be reached upon normal termination of the condition or subsequent to interruption of the evaluation due to syntax errors. The transition function *T* is realized by the state machine and completely defined in the state transition. The flowchart for question construct assessment and evaluation of the conditions and execution of the finite state automaton is depicted in Figure 3.

Our implementation of the FSA for handheld computers also considers *backtracking*. Participants can always go back to correct a mistakenly chosen answer on the previous question. Therefore, the branching to be traversed will depend upon the corrected answers.

The FSA is an efficient data structure for symbolically traversing the state space of a program. It is well suited to problem domains that possess a set of well-defined states and rules to govern state transitions [7]. To the best of our knowledge, our implementation of the FSA approach is the first of its kind.

Conclusion

Logistical constraints preclude a prolonged period of observation of volunteered subjects in a controlled environment. Handheld computers provide an opportunity for real-time monitoring in a natural setting with real cues and triggers. We need to capture drug craving, drug-use episodes, their circumstances, and measures of accompanying affect. Economic and social constraints only allow us to use the most inexpensive hardware to implement a solution.

We propose a high-level specification and a novel engineering approach that makes the use of an inexpensive low-end handheld device possible for executing moderately calculation-intensive programming code --otherwise extremely challenging-- to implement on such hardware and render a highly adaptable system. This solution enables researchers to "*branch*" their questionnaires without modifications to the source code. A question is presented to a participant if certain conditions are evaluated to *True*. The real-time evaluation of a condition provides dynamic customization to reflect participants' behaviors. This approach allows us to conduct close assessment and monitoring of symptoms, and intensity of drug withdrawal in patients.

The handheld devices are synchronized during predetermined participant visits to our clinics, and the questionnaire data are incorporated into the participants medical and research records via our larger *Human Research Information System* (*HuRIS*). This system allows us to administer adaptable assessments that are dynamically customized to reflect participants' behaviors without the need for any post-production programming with a very high level of flexibility.

Our first study uses more than 150 of such handheld devices as the primary source for the EMA studies. Outcome data, continuing to be collected, have not yet been analyzed, but preliminary data on usability are encouraging, as shown in Table 1. The favorable response from substance-abusing patients has prompted the initiation of a similar study for obesity. The results of the current research as well as those of a comprehensive usability study of the system will be reported elsewhere upon completion. This system can be seamlessly configured to provide support for research into other psychological and medical phenomena, such as eating binges, alcohol abuse or pain flare-ups.

Table 1. Data from a usability questionnaire administered to each participant two weeks after using TED in a substance-abuse study.

How would you compare the TED to answering questionnaires on paper?	
1 A lot better	43%
2 Better	39%
3 The same	11%
4 Worse	7%
5 A lot worse	0%
How do you feel about the amount of time it takes to fill out the questionnaires?	
1 It doesn't take long	82%
2 It is a little too long	18%
3 It is much too long	0%
In general, how easy is it to use the TED?	
1 Very easy	61%
2 Easy	36%
3 Difficult	3%
4 Very difficult	0%

References

[1] Tschopp M, Lovis C, Geissbuhler A. Understanding usage patterns of handheld computers in clinical practice. Proceedings of AMIA 2002, November 9-13:806-809.

[2] Stone AA, Shiffman S, Schwartz JE, Broderick JE, Hufford MR. Patient non-compliance with paper diaries. BMJ 2002;324(7347):1193-4.

[3] Laurenceau JP, Bolger N. Using diary methods to study marital and family processes. J Fam Psychol 2005; 19(1): 86-97.

[4] Vahabzadeh M, Epstein D, Mezghanni M, Lin J-L, Preston K. An electronic diary software for ecological momentary assessment (EMA) in clinical trials. Proc. 17th IEEE Symposium on Computer-Based Medical Systems (CBMS 2004) June 2004:167-172.

[5] Carroll J, Long D. Theory of finite automata. Englewood Cliffs, NJ.: Prentice Hall; 1989.

[6] Hopcroft JE, Motwani R, Ullman JD. Introduction to automata theory, languages and computation. 2nd ed. Boston, MA: Addison Wesley; 2000.

[7] Schüle T, Schneider K. Exact runtime analysis using automata-based symbolic simulation. Proc. 1st ACM & IEEE International Conference on Formal Methods and Models for Co-Design (MEMOCODE 2003) 2003:153-162.